

MORRISON & FOERSTER LLP
MICHAEL A. JACOBS (Bar No. 111664)
mjacobs@mofo.com
KENNETH A. KUWAYTI (Bar No. 145384)
kkuwayti@mofo.com
MARC DAVID PETERS (Bar No. 211725)
mdpeters@mofo.com
DANIEL P. MUINO (Bar No. 209624)
dmuino@mofo.com
755 Page Mill Road, Palo Alto, CA 94304-1018
Telephone: (650) 813-5600 / Facsimile: (650) 494-0792

BOIES, SCHILLER & FLEXNER LLP
DAVID BOIES (Admitted *Pro Hac Vice*)
dboies@bsflp.com
333 Main Street, Armonk, NY 10504
Telephone: (914) 749-8200 / Facsimile: (914) 749-8300
STEVEN C. HOLTZMAN (Bar No. 144177)
sholtzman@bsflp.com
1999 Harrison St., Suite 900, Oakland, CA 94612
Telephone: (510) 874-1000 / Facsimile: (510) 874-1460

ORACLE CORPORATION
DORIAN DALEY (Bar No. 129049)
dorian.daley@oracle.com
DEBORAH K. MILLER (Bar No. 95527)
deborah.miller@oracle.com
MATTHEW M. SARBORARIA (Bar No. 211600)
matthew.sarboraria@oracle.com
500 Oracle Parkway, Redwood City, CA 94065
Telephone: (650) 506-5200 / Facsimile: (650) 506-7114

Attorneys for Plaintiff
ORACLE AMERICA, INC.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE INC.

Defendant.

Case No. CV 10-03561 WHA

**ORACLE'S OPPOSITION TO
GOOGLE'S MOTION FOR RULE
50(b) JUDGMENT AS A MATTER
OF LAW OR FOR A NEW TRIAL**

Date: August 23, 2012
Time: 8:00 a.m.
Dept.: Courtroom 8, 19th Floor
Judge: Honorable William H. Alsup

1 **I. INTRODUCTION**

2 The Court should deny Google’s motion for judgment as a matter of law (“JMOL”), or, in
3 the alternative, for a new trial. As Google states at the outset, Google filed its motion solely for
4 the purpose of preserving its rights on appeal. Mot. at 1. Google’s motion fails to identify factual
5 or legal grounds that would be sufficient to grant judgment to Google or to *reverse* the Court’s
6 granting of Oracle’s motion for JMOL of copyright infringement for Google’s copying of the
7 decompiled files. Nor has Google shown that it is entitled to a new trial with respect to its
8 copying of the rangeCheck method.

9 Google’s theory with respect to “works as a whole” is legally incorrect and would
10 swallow copyright protection for any content—regardless of its originality or independent
11 significance—that happened to be packaged with a large amount of other work. That is
12 particularly true here, where Google would argue that copying of even very large amounts of
13 code is insignificant compared with a 15-million-line total.

14 Not only did Oracle present sufficient evidence for a reasonable jury to conclude that
15 Google literally copied significant material from Arrays.java and eight decompiled files, but the
16 clear weight of the evidence is on Oracle’s side. The parties have briefed these issues
17 extensively, and Google’s brief adds nothing new. To the extent that Google’s motion
18 incorporates Google’s previous briefs on these issues by reference, Oracle also incorporates its
19 previous briefs. *See* ECF Nos. 956, 986, 1013, and 1093, incorporated by reference herein.

20 **II. THE COURT PROPERLY INSTRUCTED THE JURY TO COMPARE**
21 **INDIVIDUAL FILES WHEN EVALUATING GOOGLE’S LITERAL**
22 **COPYING**

23 Each of the files from which Google copied—in part or in their entirety—is a complete
24 work for purposes of Google’s *de minimis* defense. Google argues that the entire Java platform is
25 necessarily the “work as a whole” because that is what Oracle registered with the Copyright
26 Office. Courts have soundly rejected that argument. The Ninth Circuit has held: “A creative
27 work does not deserve less copyright protection just because it is part of a composite work.”
28 *Hustler Mag., Inc. v. Moral Majority, Inc.*, 796 F.2d 1148, 1155 (9th Cir. 1986) (finding that a

single page in a 154-page magazine can constitute an entire work). As explained in *Los Angeles Times v. Free Republic*:

Defendants contend that plaintiffs' "work" is the entire daily newspaper because their copyright registration covers the paper as a whole rather than any particular article. Thus, they assert, copying an individual article constitutes reproduction of only a small portion of the entire work. This proposition is not supported by the case law. See *Texaco, supra*, 60 F.3d at 925-26 (copying an entire article from a journal where the copyright registration covered the journal as a whole constituted a copying of the entire work); *Hustler Magazine, supra*, 796 F.2d at 1155 (finding that "[a] creative work does not deserve less copyright protection just because it is part of a composite work" and holding that the copying of a one-page parody from a 154-page magazine constituted a copying of the entire work); *Netcom On-Line II, supra*, 923 F.Supp. at 1247 ("although many of Hubbard's lectures, policy statements, and course packets are collected into larger volumes, and registered as a whole, they may still constitute separate works for the purposes of this factor"); *Lerma, supra*, 1996 WL 633131 at *9 ("we find that the Works at issue in this case are combined in 'collections' and that each subpart must be considered a 'single work' for the purposes of fair use analysis").

L.A. Times v. Free Republic, No. CV 98-7840-MMM(AJWx), 1999 WL 33644483, at *19 (C.D. Cal. Nov. 8, 1999); see also 37 C.F.R. 202.3(b)(4)(i)(A); *Am. Geophysical Union v. Texaco Inc.*, 802 F. Supp. 1, 17 (S.D.N.Y. 1992) (holding that each article within a journal was protected by a copyright even though the publisher chose to register only each issue of the journal with the Copyright Office); *Religious Tech. Ctr. v. Lerma*, No. 95-1107-A, 1996 U.S. Dist. LEXIS 15454, at *27 (E.D. Va. Oct. 4, 1996) ("Although Lerma did not post the entirety of [the materials registered with the Copyright Office], he did post the entirety of certain discrete subparts of these series. Under the Code of Federal Regulations and under case law, these subparts constitute single works and are the benchmark against which to compare Lerma's actions.").

Moreover, copyright regulations permit the registration of multiple works on a single application. "For the purpose of registration on a single application and upon payment of a single registration fee, the following shall be considered a single work: (A) In the case of published works: all copyrightable elements that are otherwise recognizable as self-contained works, that are included in a single unit of publication, and in which the copyright claimant is the same" 37 C.F.R. § 202.3(b)(4)(i)(A). Thus when Google argues that "[t]here is no proper legal or evidentiary basis" on which Oracle's copyright registrations "can be subdivided, file-by-file,"

Google is simply wrong. Mot. at 2. Each source code file in the Java platform is an original, copyrightable work that is recognizable as a self-contained work. There was no requirement for Oracle or Sun to register each file separately. To hold otherwise would either create huge administrative burdens on the Copyright Office or permit copyists and plagiarists to steal files from large software projects with impunity, so long as they confined their theft to a small number of files. That makes no sense.

Thus the Court correctly instructed the jury that “[f]or purposes of Question No. 3, the ‘work as a whole’ is the compilable code for the individual file” ECF No. 1018 at 14-15.

III. A REASONABLE JURY COULD FIND THAT GOOGLE’S COPYING OF RANGECHECK WAS NOT DE MINIMIS

Copying is *de minimis* only “if it is so meager and fragmentary that compared to the work as a whole the average audience would not recognize the appropriation.” ECF No. 1018, JI 28; *Fisher v. Dees*, 794 F.2d 432, 434 n.2 (9th Cir. 1986). Here, Google’s appropriation of the rangeCheck code is immediately recognizable. Dr. Mitchell testified that the Java and Android versions of rangeCheck were “strikingly similar” and added:

They’re really identical up to details that don’t really matter as far as causing the method to work properly. One of the things that I just found really unusual and unexpected is the spacing around the plus signs, which seems kinds of arbitrary. You could type the spaces as you like and there are some places where the spaces are -- where there’s spacing around the plus and some not. It just seems unlikely that anybody would do that twice by accident. Really looks like -- I don’t know how this could happen except by copying the code.

RT 1255:5-15 (Mitchell).

Courts find a use *de minimis* only if it is both quantitatively and qualitatively insignificant. “Substantiality is measured by considering the qualitative and quantitative significance of the copied portion in relation to the plaintiff’s work as a whole.” *Newton v. Diamond*, 388 F.3d 1189, 1195 (9th Cir. 2004); *see also Merch. Transaction Sys., Inc. v. Nelcela, Inc.*, No. CV 02-1954-PHX-MHM, 2009 U.S. Dist. LEXIS 25663, at *61 (D. Ariz. Mar. 17, 2009) (“Thus, Nelcela will not escape liability unless it can show that the protectable elements in the Lexcel software constitute an insignificant (quantitatively and qualitatively) portion or aspect of the Lexcel software.”).

1 Courts have found that a quantitatively small taking can still support a finding of
2 infringement. In *Baxter*, the Ninth Circuit upheld denial of summary judgment for the defendant
3 even assuming the similarity between the two musical works could be reduced to a six-note
4 sequence, citing a long line of cases finding substantial similarity even where the copied portion
5 was a very small fraction of the work as a whole. *Baxter v. MCA, Inc.*, 812 F.2d 421, 425 (9th
6 Cir. 1987); *Fred Fisher, Inc. v. Dillingham*, 298 F. 145, 148 (S.D.N.Y. 1924) (Hand, J.) (eight
7 note “ostinato” held to infringe copyright in song); *see also Harper & Row, Publishers, Inc. v.*
8 *Nation Enters.*, 471 U.S. 539, 565-66, 579 (1985) (holding in fair use context that copying of 300
9 words from 200,000 word manuscript of presidential memoir was substantial).

10 Here, Oracle presented evidence that Google literally copied a quantitatively and
11 qualitatively significant portion of *Arrays.java*. That evidence was more than sufficient for a
12 reasonable juror to conclude that the copied code was not “so meager and fragmentary” as to be
13 unrecognizable as an appropriation.

14 The evidence established the qualitative significance of the copied *rangeCheck* method.
15 *rangeCheck* is included in one of the 37 API packages at issue. RT 1254:19-1255:2 (Mitchell).
16 The *rangeCheck* method operates on Android mobile devices, and the *rangeCheck* “code appears
17 in the source code archive of Samsung,” indicating that it “very likely appears on the Samsung
18 phones.” RT 1255:22-25, 1264:19-23 (Mitchell). Professor Mitchell testified that the
19 *rangeCheck* method is qualitatively significant and “useful” to Android as part of the API
20 libraries. RT 1316:17-19 (Mitchell). He testified that he analyzed the significance of *rangeCheck*
21 to other code in the same class file and found “a number of other source code [sic] in other files”
22 that called upon it. RT 1329:9-14 (Mitchell).

23 Google’s attempt to trivialize the *rangeCheck* method relies on a partial quote from
24 Professor Mitchell’s testimony to alter its meaning. ECF No. 984 at 5 (“Dr. Mitchell conceded
25 that ‘a good high school programmer’ could write *rangeCheck*.”). But Professor Mitchell actually
26 testified that “a good high school programmer or graduate student, *if told exactly what was*
27 *needed*, could write the code.” RT 1316:24-25 (Mitchell) (emphasis added). In fact, the
28 *rangeCheck* “code has some subtlety” and “the interesting part is figuring out exactly what you

1 wanted the function to do, more than realizing that function in Java code once that's understood."
 2 RT 1317:1-5 (Mitchell). rangeCheck's importance to the Android TimSort code was underscored
 3 by Dr. Bloch's testimony that "it was probably the first thing I put in there":

4 Q. Okay. Do you recall specifically when you added the -- what point in the
 5 process you added the rangeCheck function to the Timsort file?

6 A. I would assume as soon as I started writing it, you know, basically when I was
 7 going to write Timsort because I knew that it was headed for inclusion in
 arrays.java. You know, it was part of the scaffolding, so I think it was probably the
 first thing I put in there.

8 RT 821:21-822:3 (Bloch).

9 To evaluate the significance of the rangeCheck method to Android, Professor Mitchell
 10 experimented with an Android device and found that it called the rangeCheck method no less than
 11 2,600 times during start up alone. RT 1329:9-21 (Mitchell). He characterized that as "a pretty
 12 big number for the number of calls to this function." RT 1329:20-21 (Mitchell). Within
 13 Arrays.java, moreover, other methods call rangeCheck 18 times. TX 623.1. Google presented no
 14 evidence rebutting any of this testimony. Although Google *argues* in its motion that the number
 15 of times code is called is not necessarily significant (*see* Mot. at 4), Google cites *no trial evidence*
 16 and *no legal authority* to support that argument. Regardless of the proper frame of reference for
 17 the "work as a whole," the jury could have reasonably found that rangeCheck was qualitatively
 18 and quantitatively significant.

19 Google's argument that rangeCheck must be insignificant since Google removed it from
 20 Android (Mot. at 4) is factually incorrect. Even after Oracle sued, Google did not remove
 21 rangeCheck from preexisting versions of Android and continued distributing rangeCheck during
 22 trial. *See* RT 1832:3-10 (Bornstein). A reasonable jury could disregard Google's protestations
 23 that its deliberate copying was not significant.

24 Even if the Java platform were taken as the "work as a whole," Google's copying of
 25 rangeCheck would still be copyright infringement. "No plagiarist can excuse the wrong by
 26 showing how much of his work he did not pirate." *Shaw v. Lindheim*, 919 F.2d 1353, 1362 (9th
 27 Cir. 1990) (quoting 4 Nimmer on Copyright § 13.03[B][1][a)); *Sheldon v. MGM*, 81 F.2d 49, 56
 28 (2d Cir. 1936) (Hand, J.) (stating the same). Google relies solely on *Newton v. Diamond* to

support its *de minimis* argument, but the facts of *Newton* are distinguishable. Mot. at 3-4. *Newton* was a music sampling case, in which the defendant, who had copied a three-note sequence from a recording, had a license to the recording of the performance but not the composition. Thus the question in that case was whether the average audience—presumably untrained music listeners—could discern the plaintiff’s “hand as a composer, apart from his talent as a performer, from [the defendants’] use of the sample.” *Newton*, 388 F.3d at 1196. In this case, the “average audience” for the source code is (in Google’s words) people who “can read and understand code. It would include programmers and app developers.” RT 2688:12-14 (Van Nest). Such programmers and developers would still recognize Google’s copying of `rangeCheck` even if the “work as a whole” were the entire source code base for the J2SE platform. Uncontradicted testimony established that Oracle’s `rangeCheck` method had “unusual,” “unexpected,” and “arbitrary” characteristics, and that Google’s appropriation of the code was “striking.” RT 1255:5-15 (Mitchell). Thus there was sufficient evidence at trial to support the jury’s verdict of copyright infringement.

IV. GOOGLE COMITTED COPYRIGHT INFRINGEMENT BY COPYING EIGHT FILES THROUGH DECOMPIlation

On May 11, 2012, the Court granted judgment of copyright infringement as a matter of law for Google’s copying of eight entire Oracle Java files through decompilation:

The evidence at trial showed that Google decompiled eight Java files and copied them each in their entirety. No reasonable jury could find that the copying of entire computer files was *de minimis*. The trial record contains the source code for the Java code files (TX 623.2–623.8), decompiled versions of Java code files (TX 896.1–896.8), and corresponding Android code files (TX 1031–40). Professor John Mitchell testified about the decompilation process, how he determined that the eight files were decompiled and how, in a side-by-side comparison he found “that the actual code matches completely” (Tr. at 1259–1260).

Order Granting Motion for Judgment as a Matter of Law on Decompiled Files, ECF No. 1123 at 1.

In two sentences tacked on almost as an afterthought in its brief, Google asks the Court to reverse its decision and grant judgment in Google’s favor, or to grant a new trial. Google does not advance any facts or argument to support its request, except to refer to its previous briefing on

1 the subject. Mot. at 5. The Court has already considered those arguments and rejected them.
 2 The Court should reject Google's renewed motion as well.

3 **V. GOOGLE IS NOT ENTITLED TO A NEW TRIAL**

4 Although a new trial may be appropriate if "the verdict is against the weight of the
 5 evidence," *Molski v. M.J. Cable, Inc.*, 481 F.3d 724, 729 (9th Cir. 2007), Google makes no new
 6 arguments supporting a new trial but simply argues that it is entitled to a new trial "[f]or all the
 7 reasons Google is entitled to JMOL . . . related to the rangeCheck function." Mot. at 5. Google
 8 has not shown any legal errors regarding rangeCheck, nor has Google shown that the jury's
 9 verdict was against the weight of the evidence. Because Google is not entitled to JMOL for the
 10 reasons discussed above, Google is not entitled to a new trial.

11 **VI. CONCLUSION**

12 For the foregoing reasons, the Court should deny Google's motion for judgment as a
 13 matter of law, or, in the alternative, a new trial.

14
 15 Dated: July 31, 2012

MORRISON & FOERSTER LLP

17 By: /s/ Michael A. Jacobs

18 Michael A. Jacobs

19 *Attorneys for Plaintiff*
 20 ORACLE AMERICA, INC.